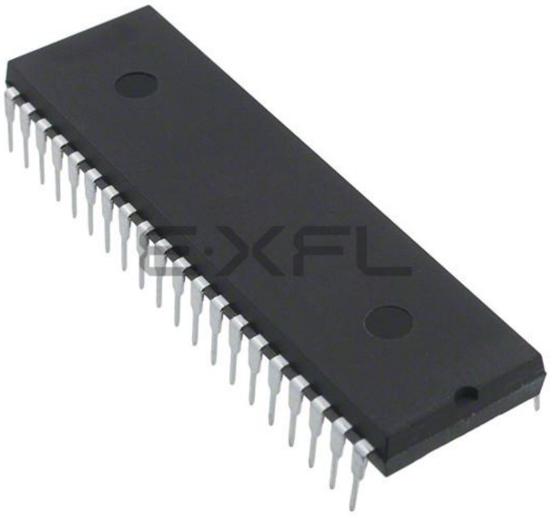


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	33
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gp32cpe

Email: info@E-XFL.COM Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

List of Chapters

Chapter 11
Low-Voltage Inhibit (LVI)

Chapter 12
Input/Output (I/O) Ports

Chapter 18
Development Support

Chapter 19
Electrical Specifications

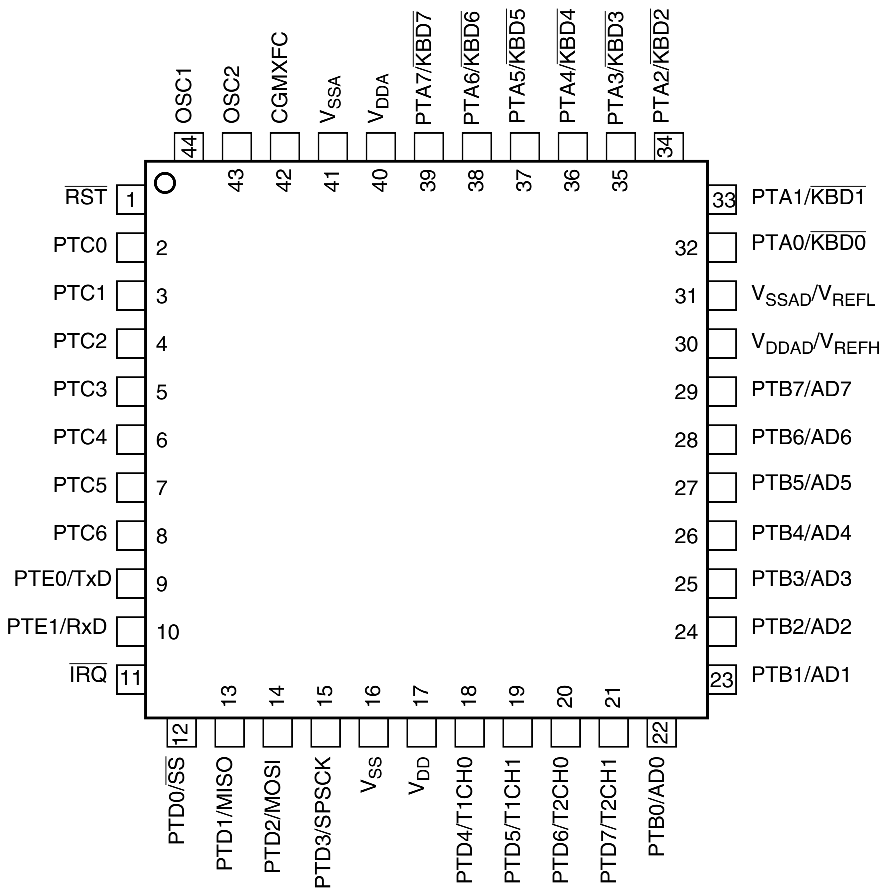


Figure 1-4. 44-Pin QFP Pin Assignments

1.5 Pin Functions

Descriptions of the pin functions are provided here.

1.5.1 Power Supply Pins (V_{DD} and V_{SS})

V_{DD} and V_{SS} are the power supply and ground pins. The MCU operates from a single power supply.

Fast signal transitions on MCU pins place high, short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as Figure 1-5 shows. Place the C1 bypass capacitor as close to the MCU as possible. Use a high-frequency-response ceramic capacitor for C1. C2 is an optional bulk current bypass capacitor for use in applications that require the port pins to source high current levels.

Chapter 2
Memory

2.1 Introduction

The CPU08 can address 64 Kbytes of memory space. The memory map, shown in Figure 2-1, includes:

- 32,256 bytes of user FLASH memory
- 512 bytes of random-access memory (RAM)
- 36 bytes of user-defined vectors
- 307 bytes of monitor ROM

2.2 Unimplemented Memory Locations

Accessing an unimplemented location can cause an illegal address reset. In the memory map (Figure 2-1) and in register figures in this document, unimplemented locations are shaded.

2.3 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on MCU operation. In the Figure 2-1 and in register figures in this document, reserved locations are marked with the word Reserved or with the letter R.

2.4 Input/Output (I/O) Section

Most of the control, status, and data registers are in the zero page area of $0000–$003F. Additional I/O registers have these addresses:

- $FE00; SIM break status register, SBSR
- $FE01; SIM reset status register, SRSR
- $FE02; reserved, SUBAR
- $FE03; SIM break flag control register, SBFCR
- $FE04; interrupt status register 1, INT1
- $FE05; interrupt status register 2, INT2
- $FE06; interrupt status register 3, INT3
- $FE07; reserved
- $FE08; FLASH control register, FLCR
- $FE09; break address register high, BRKH
- $FE0A; break address register low, BRKL
- $FE0B; break status and control register, BRKSCR
- $FE0C; LVI status register, LVISR
- $FF7E; FLASH block protect register, FLBPR
- $FFFF; COP control register, COPCTL

Data registers are shown in Figure 2-2. Table 2-1 is a list of vector locations.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE08	FLASH Control Register (FLCR)	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE09	Break Address Register High (BRKH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0A	Break Address Register Low (BRKL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0B	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0C	LVI Status Register (LVISR)	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FF7E	FLASH Block Protect Register (FLBPR)†	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	U	U	U	U	U	U	U	U
$FFFF	COP Control Register (COPCTL)	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	Unaffected by reset							

† Non-volatile FLASH register

(shaded) = Unimplemented   R = Reserved   U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 6 of 6)

Chapter 3
Low-Power Modes

3.1 Introduction

The MCU may enter two low-power modes: wait mode and stop mode. They are common to all HC08 MCUs and are entered through instruction execution. This section describes how each module acts in the low-power modes.

3.1.1 Wait Mode

The WAIT instruction puts the MCU in a low-power standby mode in which the CPU clock is disabled but the bus clock continues to run. Power consumption can be further reduced by disabling the LVI module and/or the timebase module through bits in the CONFIG register. (See Chapter 6 Configuration Register (CONFIG).)

3.1.2 Stop Mode

Stop mode is entered when a STOP instruction is executed. The CPU clock is disabled and the bus clock is disabled if the OSCSTOPENB bit in the CONFIG register is at a logic 0. (See Chapter 6 Configuration Register (CONFIG).)

3.2 Analog-to-Digital Converter (ADC)

3.2.1 Wait Mode

The ADC continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting ADCH4–ADCH0 bits in the ADC status and control register before executing the WAIT instruction.

3.2.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode after an external interrupt. Allow one conversion cycle to stabilize the analog circuitry.

3.3 Break Module (BRK)

3.3.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if the SBSW bit in the break status register is set.

5.8 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

5.8.1 Acquisition/Lock Time Definitions

Typical control systems refer to the acquisition time or lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL, the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition, regardless of the size of the step input. For example, consider a system with a 5 percent acquisition time tolerance. If a command instructs the system to change from 0 Hz to 1 MHz, the acquisition time is the time taken for the frequency to reach 1 MHz ±50 kHz. Fifty kHz = 5% of the 1-MHz step input. If the system is operating at 1 MHz and suffers a –100-kHz noise hit, the acquisition time is the time taken to return from 900 kHz to 1 MHz ±5 kHz. Five kHz = 5% of the 100-kHz step input.

Other systems refer to acquisition and lock times as the time the system takes to reduce the error between the actual output and the desired output to within specified tolerances. Therefore, the acquisition or lock time varies according to the original error in the output. Minor errors may not even be registered. Typical PLL applications prefer to use this definition because the system requires the output frequency to be within a certain tolerance of the desired frequency regardless of the size of the initial error.

5.8.2 Parametric Influences on Reaction Time

Acquisition and lock times are designed to be as short as possible while still providing the highest possible stability. These reaction times are not constant, however. Many factors directly and indirectly affect the acquisition time.

The most critical parameter which affects the reaction times of the PLL is the reference frequency, f_{RDV}. This frequency is the input to the phase detector and controls how often the PLL makes corrections. For stability, the corrections must be small compared to the desired frequency, so several corrections are required to reduce the frequency error. Therefore, the slower the reference the longer it takes to make these corrections. This parameter is under user control via the choice of crystal frequency f_{XCLK} and the R value programmed in the reference divider. (See 5.3.3 PLL Circuits, 5.3.6 Programming the PLL, and 5.5.6 PLL Reference Divider Select Register.)

Another critical parameter is the external filter network. The PLL modifies the voltage on the VCO by adding or subtracting charge from capacitors in this network. Therefore, the rate at which the voltage changes for a given frequency error (thus change in charge) is proportional to the capacitance. The size of the capacitor also is related to the stability of the PLL. If the capacitor is too small, the PLL cannot make small enough adjustments to the voltage and the system cannot lock. If the capacitor is too large, the PLL may not be able to adjust the voltage in a reasonable time. (See 5.8.3 Choosing a Filter.)

Also important is the operating voltage potential applied to V_{DDA}. The power supply potential alters the characteristics of the PLL. A fixed value is best. Variable supplies, such as batteries, are acceptable if they vary within a known range at very slow speeds. Noise on the power supply is not acceptable, because it causes small frequency errors which continually change the acquisition time of the PLL.

The COP counter is a free-running 6-bit counter preceded by a 12-bit prescaler counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after 262,128 or 8176 CGMXCLK cycles, depending on the state of the COP rate select bit, COPRS, in the configuration register. With a 8176 CGMXCLK cycle overflow option, a 32.768-kHz crystal gives a COP timeout period of 250 ms. Writing any value to location $FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12 through 5 of the prescaler.

NOTE

Service the COP immediately after reset and before entering or after exiting stop mode to guarantee the maximum time before the first COP counter overflow.

A COP reset pulls the $\overline{\text{RST}}$ pin low for 32 CGMXCLK cycles and sets the COP bit in the reset status register (RSR).

In monitor mode, the COP is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ}}$ is held at V_{TST}. During the break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP.

NOTE

Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.

7.3 I/O Signals

The following paragraphs describe the signals shown in Figure 7-1.

7.3.1 CGMXCLK

CGMXCLK is the crystal oscillator output signal. CGMXCLK frequency is equal to the crystal frequency.

7.3.2 STOP Instruction

The STOP instruction clears the COP prescaler.

7.3.3 COPCTL Write

Writing any value to the COP control register (COPCTL) (see 7.4 COP Control Register) clears the COP counter and clears bits 12 through 5 of the prescaler. Reading the COP control register returns the low byte of the reset vector.

7.3.4 Power-On Reset

The power-on reset (POR) circuit clears the COP prescaler 4096 CGMXCLK cycles after power-up.

7.3.5 Internal Reset

An internal reset clears the COP prescaler and the COP counter.

7.3.6 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the COP prescaler.

11.3.3 Voltage Hysteresis Protection

Once the LVI has triggered (by having V_{DD} fall below V_{TRIPF}), the LVI will maintain a reset condition until V_{DD} rises above the rising trip point voltage, V_{TRIPR}. This prevents a condition in which the MCU is continually entering and exiting reset if V_{DD} is approximately equal to V_{TRIPF}. V_{TRIPR} is greater than V_{TRIPF} by the hysteresis voltage, V_{HYS}.

11.3.4 LVI Trip Selection

The LVI5OR3 bit in the configuration register selects whether the LVI is configured for 5-V or 3-V protection.

NOTE

The microcontroller is guaranteed to operate at a minimum supply voltage. The trip point (V_{TRIPF} [5 V] or V_{TRIPF} [3 V]) may be lower than this. (See Chapter 19 Electrical Specifications for the actual trip point voltages.)

11.4 LVI Status Register

The LVI status register (LVISR) indicates if the V_{DD} voltage was detected below the V_{TRIPF} level.

Address: $FE0C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LVIOUT	0	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-3. LVI Status Register (LVISR)

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} voltage falls below the V_{TRIPF} trip voltage. (See Table 11-1.) Reset clears the LVIOUT bit.

Table 11-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

11.5 LVI Interrupts

The LVI module does not generate interrupt requests.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0004	Data Direction Register A (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC)	Read:	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE)	Read:	0	0	0	0	0	0	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
$000C	Data Direction Register E (DDRE)	Read:	0	0	0	0	0	0	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Port A Input Pullup Enable Register (PTAPUE)	Read:	PTAPUE7	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E	Port C Input Pullup Enable Register (PTCPUE)	Read:	0	PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1	PTCPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000F	Port D Input Pullup Enable Register (PTDPUE)	Read:	PTDPUE7	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1	PTDPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 12-1. I/O Port Register Summary (Continued)

Table 12-1. Port Control Register Bits Summary

<table>
<tr><th>Port</th><th>Bit</th><th>DDR</th><th colspan="2">Module Control</th><th>Pin</th></tr>
<tr><td rowspan="8">A</td><td>0</td><td>DDRA0</td><td rowspan="8">KBD</td><td>KBIE0</td><td>PTA0/KBD0</td></tr>
<tr><td>1</td><td>DDRA1</td><td>KBIE1</td><td>PTA1/KBD1</td></tr>
<tr><td>2</td><td>DDRA2</td><td>KBIE2</td><td>PTA2/KBD2</td></tr>
<tr><td>3</td><td>DDRA3</td><td>KBIE3</td><td>PTA3/KBD3</td></tr>
<tr><td>4</td><td>DDRA4</td><td>KBIE4</td><td>PTA4/KBD4</td></tr>
<tr><td>5</td><td>DDRA5</td><td>KBIE5</td><td>PTA5/KBD5</td></tr>
<tr><td>6</td><td>DDRA6</td><td>KBIE6</td><td>PTA6/KBD6</td></tr>
<tr><td>7</td><td>DDRA7</td><td>KBIE7</td><td>PTA7/KBD7</td></tr>
<tr><td rowspan="8">B</td><td>0</td><td>DDRB0</td><td rowspan="8">ADC</td><td rowspan="8">ADCH4–ADCH0</td><td>PTB0/AD0</td></tr>
<tr><td>1</td><td>DDRB1</td><td>PTB1/AD1</td></tr>
<tr><td>2</td><td>DDRB2</td><td>PTB2/AD2</td></tr>
<tr><td>3</td><td>DDRB3</td><td>PTB3/AD3</td></tr>
<tr><td>4</td><td>DDRB4</td><td>PTB4/AD4</td></tr>
<tr><td>5</td><td>DDRB5</td><td>PTB5/AD5</td></tr>
<tr><td>6</td><td>DDRB6</td><td>PTB6/AD6</td></tr>
<tr><td>7</td><td>DDRB7</td><td>PTB7/AD7</td></tr>
<tr><td rowspan="7">C</td><td>0</td><td>DDRC0</td><td rowspan="7" colspan="2"></td><td>PTC0</td></tr>
<tr><td>1</td><td>DDRC1</td><td>PTC1</td></tr>
<tr><td>2</td><td>DDRC2</td><td>PTC2</td></tr>
<tr><td>3</td><td>DDRC3</td><td>PTC3</td></tr>
<tr><td>4</td><td>DDRC4</td><td>PTC4</td></tr>
<tr><td>5</td><td>DDRC5</td><td>PTC5</td></tr>
<tr><td>6</td><td>DDRC6</td><td>PTC6</td></tr>
<tr><td rowspan="8">D</td><td>0</td><td>DDRD0</td><td rowspan="4">SPI</td><td rowspan="4">SPE</td><td>PTD0/$\overline{SS}$</td></tr>
<tr><td>1</td><td>DDRD1</td><td>PTD1/MISO</td></tr>
<tr><td>2</td><td>DDRD2</td><td>PTD2/MOSI</td></tr>
<tr><td>3</td><td>DDRD3</td><td>PTD3/SPSCK</td></tr>
<tr><td>4</td><td>DDRD4</td><td rowspan="2">TIM1</td><td>ELS0B:ELS0A</td><td>PTD4/T1CH0</td></tr>
<tr><td>5</td><td>DDRD5</td><td>ELS1B:ELS1A</td><td>PTD5/T1CH1</td></tr>
<tr><td>6</td><td>DDRD6</td><td rowspan="2">TIM2</td><td>ELS0B:ELS0A</td><td>PTD6/T2CH0</td></tr>
<tr><td>7</td><td>DDRD7</td><td>ELS1B:ELS1A</td><td>PTD7/T2CH1</td></tr>
<tr><td rowspan="2">E</td><td>0</td><td>DDRE0</td><td rowspan="2">SCI</td><td rowspan="2">ENSCI</td><td>PTE0/TxD</td></tr>
<tr><td>1</td><td>DDRE1</td><td>PTE1/RxD</td></tr>
</table>

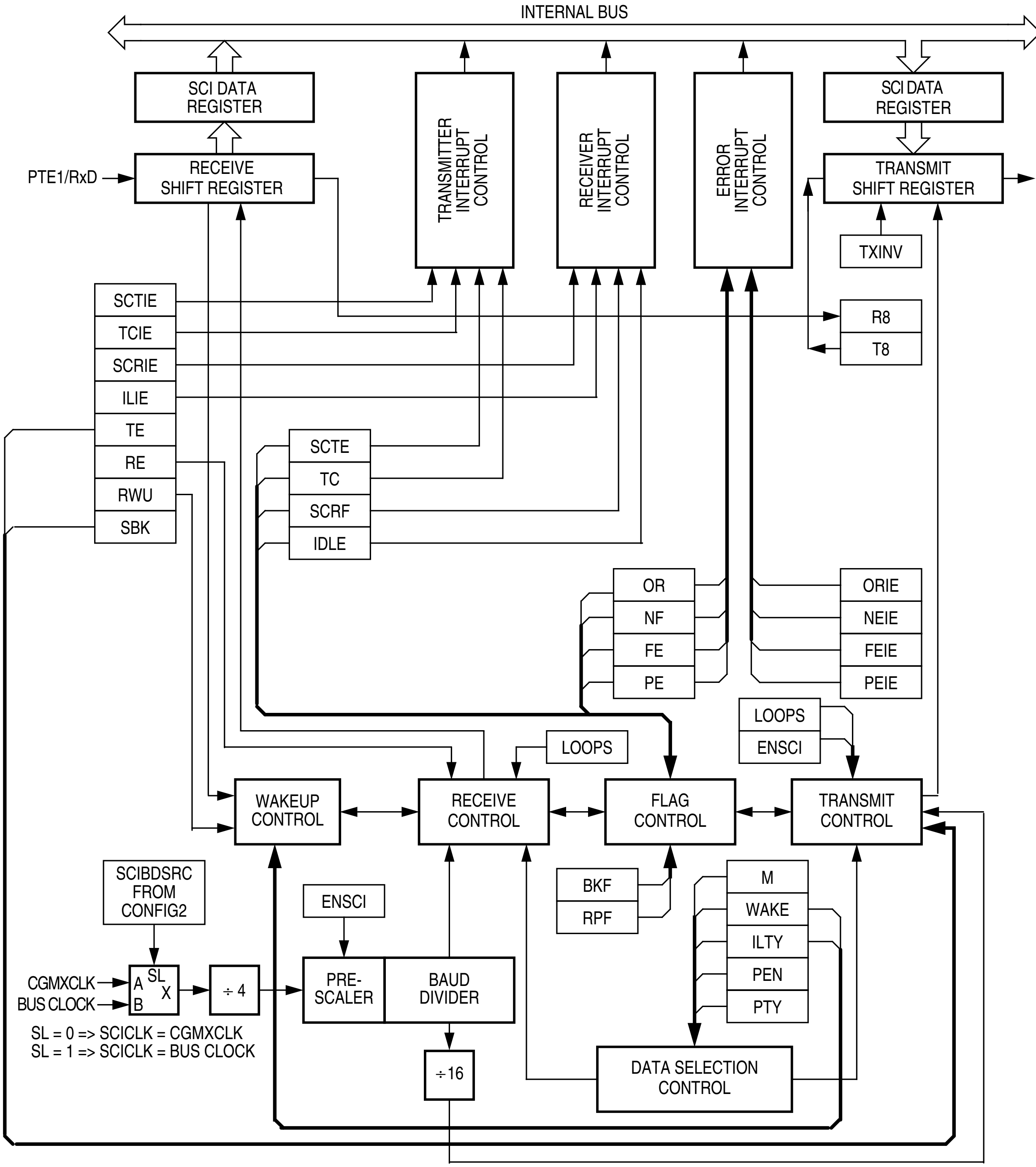


Figure 13-1. SCI Module Block Diagram

Table 13-5. Character Format Selection

Control Bits		Character Format				
M	PEN and PTY	Start Bits	Data Bits	Parity	Stop Bits	Character Length
0	0X	1	8	None	1	10 bits
1	0X	1	9	None	1	11 bits
0	10	1	7	Even	1	10 bits
0	11	1	7	Odd	1	10 bits
1	10	1	8	Even	1	11 bits
1	11	1	8	Odd	1	11 bits

13.8.2 SCI Control Register 2

SCI control register 2:

- Enables the following CPU interrupt requests:
  - Enables the SCTE bit to generate transmitter CPU interrupt requests
  - Enables the TC bit to generate transmitter CPU interrupt requests
  - Enables the SCRF bit to generate receiver CPU interrupt requests
  - Enables the IDLE bit to generate receiver CPU interrupt requests
- Enables the transmitter
- Enables the receiver
- Enables SCI wakeup
- Transmits SCI break characters

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 13-10. SCI Control Register 2 (SCC2)

SCTIE — SCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate SCI transmitter CPU interrupt requests. Reset clears the SCTIE bit.

1 = SCTE enabled to generate CPU interrupt
0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate SCI transmitter CPU interrupt requests. Reset clears the TCIE bit.

1 = TC enabled to generate CPU interrupt requests
0 = TC not enabled to generate CPU interrupt requests

SCRIE — SCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate SCI receiver CPU interrupt requests. Reset clears the SCRIE bit.

1 = SCRF enabled to generate CPU interrupt
0 = SCRF not enabled to generate CPU interrupt

14.5.1.3 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. Table 14-3 summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

Table 14-3. Interrupt Sources

Priority	Interrupt Source	Interrupt Status Register Flag
Highest	Reset	—
↑	SWI instruction	—
	$\overline{\text{IRQ}}$ pin	IF1
	PLL	IF2
	TIM1 channel 0	IF3
	TIM1 channel 1	IF4
	TIM1 overflow	IF5
	TIM2 channel 0	IF6
	TIM2 channel 1	IF7
	TIM2 overflow	IF8
	SPI receiver full	IF9
	SPI transmitter empty	IF10
	SCI receive error	IF11
	SCI receive	IF12
	SCI transmit	IF13
	Keyboard	IF14
↓	ADC conversion complete	IF15
Lowest	Timebase module	IF16

Interrupt Status Register 1

Address: $FE04

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-12. Interrupt Status Register 1 (INT1)

IF6–IF1 — Interrupt Flags 1–6

These flags indicate the presence of interrupt requests from the sources shown in Table 14-3.

1 = Interrupt request present

0 = No interrupt request present

Chapter 15
Serial Peripheral Interface Module (SPI)

15.1 Introduction

This section describes the serial peripheral interface (SPI) module, which allows full-duplex, synchronous, serial communications with peripheral devices.

15.2 Features

Features of the SPI module include:

- Full-duplex operation
- Master and slave modes
- Double-buffered operation with separate transmit and receive registers
- Four master mode frequencies (maximum = bus frequency ÷ 2)
- Maximum slave mode frequency = bus frequency
- Serial clock with programmable polarity and phase
- Two separately enabled interrupts:
  - SPRF (SPI receiver full)
  - SPTE (SPI transmitter empty)
- Mode fault error flag with CPU interrupt capability
- Overflow error flag with CPU interrupt capability
- Programmable wired-OR mode
- I/O (input/output) port bit(s) software configurable with pullup device(s) if configured as input port bit(s)

15.3 Pin Name Conventions

The text that follows describes the SPI. The SPI I/O pin names are $\overline{SS}$ (slave select), SPSCK (SPI serial clock), CGND (clock ground), MOSI (master out slave in), and MISO (master in/slave out). The SPI shares four I/O pins with four parallel I/O ports.

The full names of the SPI I/O pins are shown in Table 15-1. The generic pin names appear in the text that follows.

Table 15-1. Pin Name Conventions

SPI Generic Pin Names:		**MISO**	**MOSI**	$\overline{SS}$	**SPSCK**	**CGND**
Full SPI Pin Names:	SPI	PTD1/MISO	PTD2/MOSI	PTD0/$\overline{SS}$	PTD3/SPSCK	V_{SS}

Reading the SPI status and control register with SPRF set and then reading the receive data register clears SPRF. The clearing mechanism for the SPTE flag is always just a write to the transmit data register.

The SPI transmitter interrupt enable bit (SPTIE) enables the SPTE flag to generate transmitter CPU interrupt requests, provided that the SPI is enabled (SPE = 1).

The SPI receiver interrupt enable bit (SPRIE) enables the SPRF bit to generate receiver CPU interrupt requests, regardless of the state of the SPE bit. (See Figure 15-11.)

The error interrupt enable bit (ERRIE) enables both the MODF and OVRF bits to generate a receiver/error CPU interrupt request.

The mode fault enable bit (MODFEN) can prevent the MODF flag from being set so that only the OVRF bit is enabled by the ERRIE bit to generate receiver/error CPU interrupt requests.

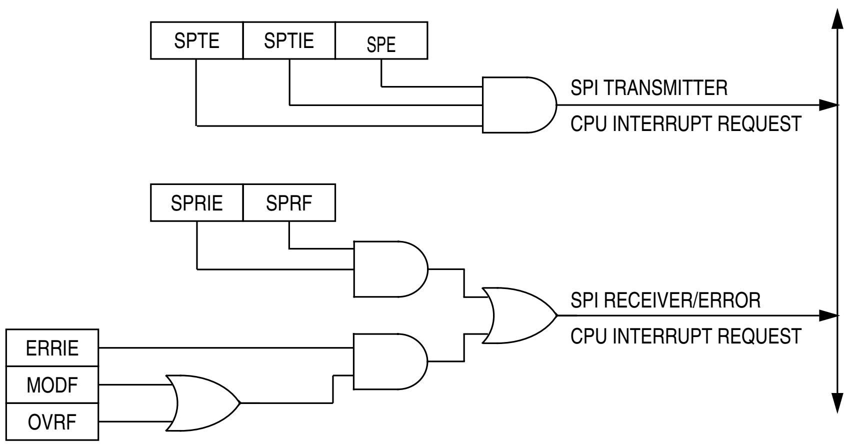


Figure 15-11. SPI Interrupt Request Generation

The following sources in the SPI status and control register can generate CPU interrupt requests:

- SPI receiver full bit (SPRF) — The SPRF bit becomes set every time a byte transfers from the shift register to the receive data register. If the SPI receiver interrupt enable bit, SPRIE, is also set, SPRF generates an SPI receiver/error CPU interrupt request.
- SPI transmitter empty (SPTE) — The SPTE bit becomes set every time a byte transfers from the transmit data register to the shift register. If the SPI transmit interrupt enable bit, SPTIE, is also set, SPTE generates an SPTE CPU interrupt request.

Chapter 16
Timebase Module (TBM)

16.1 Introduction

This section describes the timebase module (TBM). The TBM will generate periodic interrupts at user selectable rates using a counter clocked by the external crystal clock. This TBM version uses 15 divider stages, eight of which are user selectable.

16.2 Features

Features of the TBM module include:

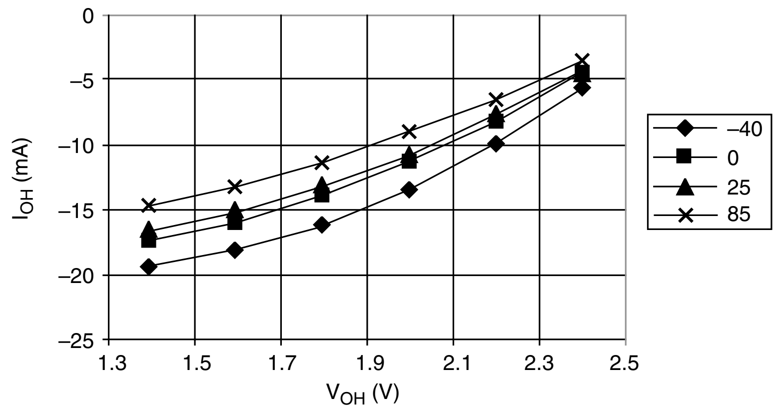
- Software programmable 1-Hz, 4-Hz, 16-Hz, 256-Hz, 512-Hz, 1024-Hz, 2048-Hz, and 4096-Hz periodic interrupt using external 32.768-kHz crystal
- Configurable for operation during stop mode to allow periodic wakeup from stop

16.3 Functional Description

NOTE

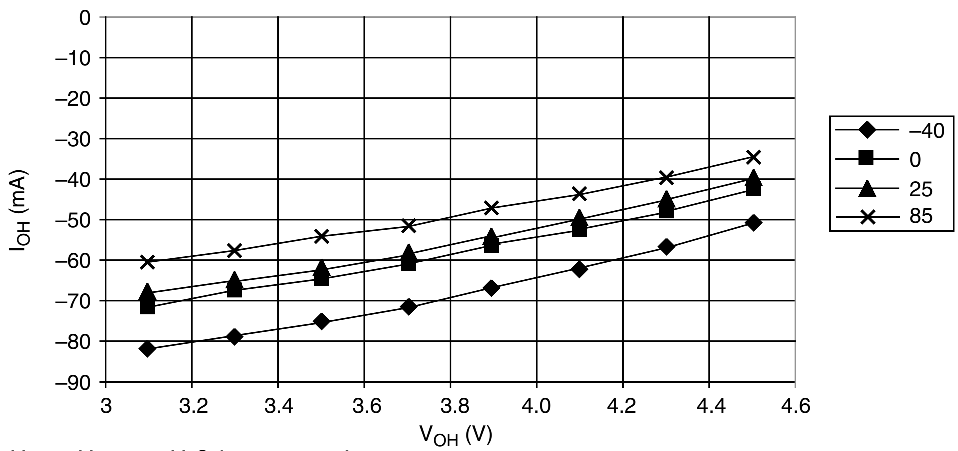
This module is designed for a 32.768-kHz oscillator.

This module can generate a periodic interrupt by dividing the crystal frequency, CGMXCLK. The counter is initialized to all 0s when TBON bit is cleared. The counter, shown in Figure 16-1, starts counting when the TBON bit is set. When the counter overflows at the tap selected by TBR2:TBR0, the TBIF bit gets set. If the TBIE bit is set, an interrupt request is sent to the CPU. The TBIF flag is cleared by writing a 1 to the TACK bit. The first time the TBIF flag is set after enabling the timebase module, the interrupt is generated at approximately half of the overflow period. Subsequent events occur at the exact period.



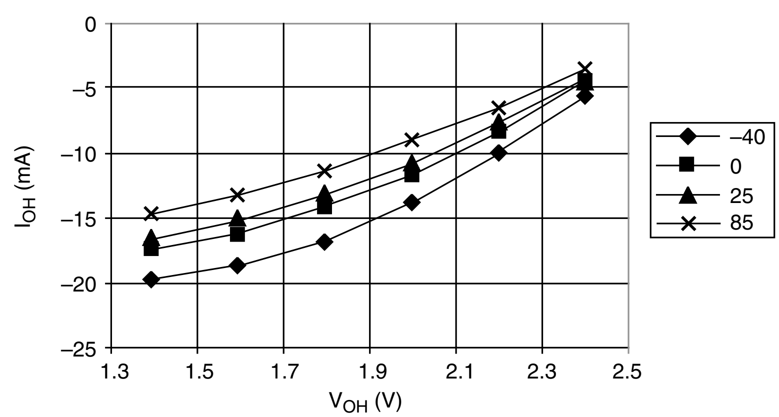
$V_{OH} > V_{DD}$ –0.5 V @ I_{OH} = –4.0 mA

Figure 19-5. Typical High-Side Driver Characteristics – Port PTC4–PTC0 (V_{DD} = 2.7 Vdc)



$V_{OH} > V_{DD}$ –0.8 V @ I_{OH} = –2.0 mA
$V_{OH} > V_{DD}$ –1.5 V @ I_{OH} = –10.0 mA

Figure 19-6. Typical High-Side Driver Characteristics – Ports PTB7–PTB0, PTC6–PTC5, PTD7–PTD0, and PTE1–PTE0 (V_{DD} = 5.5 Vdc)



$V_{OH} > V_{DD}$ –0.3 V @ I_{OH} = –0.6 mA
$V_{OH} > V_{DD}$ –1.0 V @ I_{OH} = –4.0 mA

Figure 19-7. Typical High-Side Driver Characteristics – Ports PTB7–PTB0, PTC6–PTC5, PTD7–PTD0, and PTE1–PTE0 (V_{DD} = 2.7 Vdc)